Learning Deep Visuomotor Policies for Dexterous Hand Manipulation

Divye Jain[1], Andrew Li[1], Shivam Singhal[1], Aravind Rajeswaran[1], Vikash Kumar[2], Emanuel Todorov[1,3]

[1] University of Washington [2] Google Brain [3] Roboti LLC

***Abstract*— Multi-fingered dexterous hands are versatile and capable of acquiring a diverse set of skills such as grasping, in-hand manipulation, and tool use. To fully utilize their versatility in real-world scenarios, we require algorithms and policies that can control them using on-board sensing capabilities, without relying on external tracking or motion capture systems. Cameras and tactile sensors are the most widely used on-board sensors that do not require instrumentation of the world. In this work, we demonstrate an imitation learning based approach to train deep visuomotor policies for a variety of manipulation tasks with a *simulated* five fingered dexterous hand. These policies directly control the hand using high dimensional visual observations of the world and propreoceptive observations from the robot, and can be trained efficiently with a few hundred expert demonstration trajectories. We also find that using touch sensing information enables faster learning and better asymptotic performance for tasks with high degree of occlusions. Video demonstration of our results are available at: https://sites.google.com/view/hand-vil/**

I. Introduction

For robots to be functional and competent in a wide variety of uncertain and dynamic environments, the ability to perceive the world and use rich sensory information, and the capability to influence the world through versatile manipulators, are among the most important cornerstones. Multi-fingered hands are among the most versatile manipulators and allow for a variety of contact-rich tasks such as in-hand manipulation, complex grasping, and tool usage. Similarly, visual and tactile sensing are extremely versatile, allowing to perceive and describe a wide variety of environments and scenarios. However, the versatility enabled by these manipulators and sensors comes at the steep price of high dimensional observation and action spaces, which makes the problem of controller synthesis particularly challenging. In this work, we study methods to enable learning of sensorimotor policies that are capable of complex dexterous manipulation from raw visual and tactile feedback.

In most real-world scenarios such as a home, or disaster rescue, it is often difficult or impossible to instrument the world with external motion capture or tracking systems. Thus, robotic agents must have the capability to function with only on-board sensors, and without reliance on very detailed models which may be hard to obtain for unseen scenarios. Learning based techniques provide a model-agnostic approach to synthesizing controllers. Furthermore, they often combine perception and control into a sensorimotor loop enabling each part to adapt to the capabilities of the other. Our goal in this work is to provide a proof of concept that deep sensorimotor policies can be learned for complex dexterous manipulation tasks with free objects, thereby alleviating strict requirements for explicitly constructing and estimating compact state representations.

Correspond to dpjain@cs.uw.edu, aravraj@cs.uw.edu, and vikashplus@google.com.

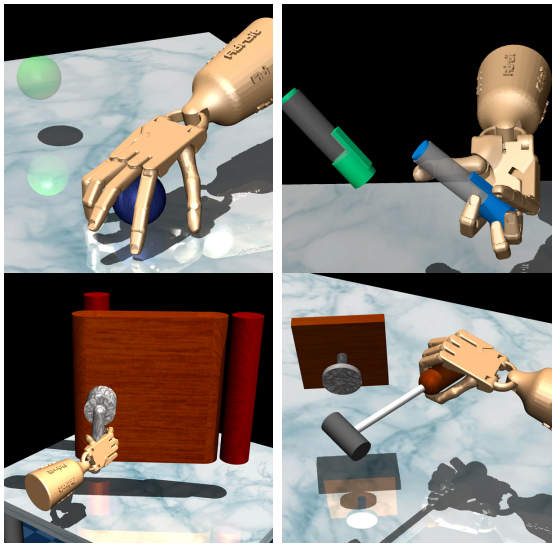


Fig. 1: Suite of dexterous hand manipulation tasks from Rajeswaran et al. [1]. The set of tasks consist of relocating an object (blue) to a target location (green), repositioning a pen to a target configuration, opening a door, and hammering a nail. Without visual information, the robot cannot locate the object or the target, and hence these tasks cannot be solved without vision. All tasks involve position control for the hand.

The deep sensorimotor policies we consider are high capacity with few hundred thousand parameters, necessitating efficient algorithms to train them. Furthermore, dexterous hand manipulation has proven to be a challenging setting with reinforcement learning [1], [2], [3], requiring many hundreds of CPU hours, even for small networks acting on compact state representations. To overcome these algorithmic and computational limitations, we employ the use of imitation learning. Of course, imitation learning requires access to demonstrations; we outline different ways to obtain them based on the problem setting in Section III.

A. Our Contributions

We consider a suite of dexterous manipulation tasks, recently introduced in Rajeswaran et al. [1], and show that deep visuomotor policies can be trained with imitation learning. These tasks require careful coordination of a large number of joints in addition to understanding the high dimensional visual observation space. The capability of visual

imitation learning to train policies with close to 100% success rate, with just a few hundred demonstrations, and under an hour of training on a workstation with 4 GPUs, is surprising and far from obvious.

In addition, as a secondary contribution, we show that combining touch sensing information with visual inputs leads to faster training and better asymptotic performance for contact-rich dexterous manipulation tasks. Although the gains are task specific, we find the gains of using touch information in addition to vision to be particularly noticeable for tasks and viewpoints with occlusions. Our results are currently in simulation. However they complement a number of recent studies (e.g. [2], [4], [5], [6], [7]) which demonstrate that policies and insights from simulation can transfer to real systems. Thus, we expect the insights from this work to both help with direct hardware training as well as accelerate approaches for simulation to reality transfer. Video demonstration of our results and code are available in the project website: https://sites.google.com/view/hand-vil/

II. Related Work

Learning control policies for robotic manipulation tasks using high dimensional sensory descriptions (such as vision) spans multiple research fields including robotics, computer vision, and machine learning. A detailed survey that spans all the fields is beyond the scope of this paper, and we provide a short survey along three primary research threads. The advances that we survey in this section are largely complementary to the research questions studied in this work. Our primary focus is to show that imitation learning approaches (as opposed to RL) provides a powerful platform to train complex dexterous manipulation policies capable of working directly with raw visual and tactile inputs.

Multi-fingered hands and dexterous manipulation: There is a large body of prior work on both design of anthropomorphic hands, as well as obtaining controllers for them. Some works have obtained successful control policies by simplifying the mechanical design [8], [9]. For more complex models and hands, trajectory optimization and policy search have produced succesful behaviors (e.g. [1], [10], [11], [12]). Robustification strategies, primarily based on ensembles of dynamics models [4], [13], have been used to successfully transfer results from simulation to hardware [13], [5], [2], [7]. Classical approaches to grasping and manipulation that reason based on geometry, stability, and force-closure [14], [15], [16], [17] as well as direct RL on the hardware [18], [19], [20] have also shown promising results in the past. However, most of these works rely on a compact state representations, while our goal is to learn visuomotor policies to overcome this limitation. Prior work has also investigated the benefit of tactile sensing for manipulation tasks and found them to be generally useful [18], [21], [22], [23].

Learning visuomotor policies: Prior work has explored a number of ways to learn visuomotor policies. Some works have concentrated on being able to transfer results from simulation to hardware – for example, using ensemble methods like domain randomization [24], [25], or progressively adapting the visual features [26]. Other works have learned visuomotor policies directly in the real world [27], [28]. Our proposed approach is complementary to the aforementioned works, in the sense that visual imitation learning can be used to accelerate (reduce compute) learning of either robust policies in simulation, or accelerate (reduce samples) real world learning of visuomotor policies.

Visual imitation learning: Imitation learning has been emerging as a promising paradigm for robotic manipulation [29], [30], [1], [31]. In particular, recent efforts have explored architectures for visual imitation learning [32] and combining it with reinforcement learning [31], [33].

We note that our goal in this work is not to propose a new imitation learning algorithm or an architecture for visuomotor policies. Rather, we explore the use of existing techniques for the problem of dexterous manipulation with multi-fingered hands. Our main contribution is to outline that in most cases where visual reinforcement leanring is being used, it is possible to replace it with visual imitation leanring with minimal additional assumptions or requirements. We find it encouraging that existing imitation learning methods already work well, and interesting future directions would include exploring alternate architectures for fusing information from multiple sensory streams.

III. Problem Formulation and Algorithm

In this work, we are interested in learning sensorimotor policies that directly map from raw sensor observations to actions such as motor torques. We model this control problem as a Partially Observed Markov Decision Process (POMDP), which is defined using the tuple:

$$\text{(POMDP)} \quad \mathcal{M} = \{\mathcal{S}, \mathcal{X}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \mathcal{G}, \rho_0, \gamma\}.$$

$\mathcal{S} \in \mathbb{R}^{n_s}$ and $\mathcal{X} \in \mathbb{R}^n$ represent the state and observation spaces respectively; the state is not directly observed and we get access to only the observations (sensory descriptions) of the world. $\mathcal{A} \in \mathbb{R}^m$ represents the action space. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents the reward function. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ and $\mathcal{G} : \mathcal{S} \times \mathcal{X} \to \mathbb{R}^+$ represent the state transition and observation models respectively. ρ_0 is the probability distribution over initial states and $\gamma \in [0, 1)$ is a discount factor. Let $\mathbf{h}_t = (\mathbf{x}_0, \mathbf{a}_0, \mathbf{x}_1, \mathbf{a}_1, \ldots \mathbf{x}_t)$ denote the history observed till time t, and let $\mathcal{H}$ represent the space of histories. We wish to solve for a policy of the form: $\pi : \mathcal{H} \times \mathcal{A} \to \mathbb{R}_+$, which optimizes the expected sum of rewards:

$$\eta(\pi) = \mathbb{E}_{\pi,\mathcal{M}}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{1}$$

Different assumptions about the various components of the POMDP give rise to different problem settings. In this work, we consider the setting where we do not have full knowledge of $\mathcal{T}$ and $\mathcal{G}$: a regime in which imitation learning shines. Furthermore, we do not attempt to learn a representation for the state explicitly, but rather learn policies that directly act based on the observations. Before discussing the algorithmic details, we present the tasks we consider in this work and their grounding in the POMDP formulation.

A. Task and Robot Description

We consider the hand manipulation suite of tasks introduced in Rajeswaran et al. [1] and depicted in Figure 1. The tasks are simulated in MuJoCo [34], and require the five-fingered Adroit hand [35] to perform a variety of manipulation behaviors like:

- **Object relocation:** grasp a sphere and move it to a target location.
- **In-hand manipulation:** reposition the pen to a desired target pose.
- **Tool usage:** pick up a hammer and use it to drive the nail into the board.
- **Door Opening:** undo a latch and open the door using the door handle.

The observations consist of all the sensing capabilities available to the robot in a real-world deployment setting. This includes a camera providing a visual description of the scene, proprioceptive sensors providing information about the configuration of the robot, and touch sensors which can sense on-off contact events. The hand is position controlled and the policy must command the desired positions (configuration) of the hand in the next timestep.

B. Imitation Learning

A variety of algorithmic solution concepts exist for solving the POMDP problem. One straightforward approach is model-free RL in the visual space, using policy gradient or actor critic algorithms [36], [37], [38], [39], [40], [41]. For the manipulation tasks we consider in this work, RL from scratch has been reported to be very inefficient even for compact state spaces [1]. It is well known that RL scales poorly with the dimensionality of the observation and action spaces [42], and our experimental findings confirm that RL in the visual space is even more inefficient compared to learning with a compact state representation. An alternate approach that is more efficient is imitation learning, which provides more stable targets for the policy to approximate while also reducing exploration burden [1], [43]. We outline few ways to obtain successful demonstraitons below:

1) ***(Access to a simulator)*** If we have a simulator, we can draw upon various optimization algorithms (using the compact representation of the simulator) to provide demonstrations. Policy and trajectory optimization have demonstrated impressive results in simulation (e.g. [44], [45], [10], [46], [3]). For complex tasks where policy or trajectory optimization is unable to find successful solutions, recent work suggests that it is possible to learn successful policies by combining RL with a small number of *human* demonstrations collected through virtual reality [1] or motion capture [47].
2) ***(Access to compact state representation)*** In many scenarios, even if state estimation is difficult at deployment time, we may have access to accurate compact states during training time. Examples include training in a well instrumented laboratory setup (with motion capture and tracking systems), and deployment for example in the home of a user. In such situations, either model-based or model-free RL can be used to train a policy much faster using the compact state representation compared to the visual representation. Demonstrations from this compact expert can be used to train a policy that directly acts on high dimensional visual observations.
3) ***(Access to human demonstrations)*** Humans can use tracking devices (e.g. *cyberglove*), and teleoperate the robot to provide demonstrations. In cases where no additional instrumentation is available, the demonstration data can be obtained through kinesthetic teaching.

In this work, we are agnostic to how the demonstrations are collected; and focus on how to use the demonstration data to learn visuomotor policies. We describe how we realized the expert for the purposes of our experiments in Section V.

Let $\boldsymbol{\pi_\theta} \in \Pi$ denote a family of policies (parameterized by $\boldsymbol{\theta}$) over which we are searching. For generality, we consider a stochastic policy that provides the conditional distribution over actions. Let $\boldsymbol{\pi^e}$ denote the expert providing the demonstrations. One approach to imitation learning is behavior cloning (or batch supervised learning), which corresponds to the following optimization problem:

$$\text{(BC)} \quad \boldsymbol{\pi_{\theta^*}} = \arg\max_{\boldsymbol{\pi_\theta} \in \Pi} \mathbb{E}_{\boldsymbol{\tau}^e} \left[\ln \boldsymbol{\pi_\theta}(\mathbf{a}_t^e | \mathbf{h}_t) \right]. \tag{2}$$

The data for the maximum likelihood estimation ($\boldsymbol{\tau}^e$) is trajectories sampled from $\mathcal{M}$ using $\boldsymbol{\pi^e}$. Each time step in the trajectory contains the data: $(\mathbf{x}_t, \mathbf{a}_t^e)$, where $\mathbf{a}_t^e$ is used to explicitly denote the action taken by the expert. Since behavior cloning requires collecting data only with $\boldsymbol{\pi^e}$, it presents a very convenient approach when a human is providing demonstrations.

In general, behavior cloning may not converge to the optimal policy due to a distribution mismatch [48]. Specifically, the data distribution used in behavior cloning ($\boldsymbol{\tau}^e$) may not match the distribution of data we get by deploying the policy (i.e. trajectories when interacting with $\mathcal{M}$ using $\boldsymbol{\pi_\theta}$). DAgger [48] address this distributional mismatch problem by solving a modified optimization problem:

$$\text{(IL)} \quad \boldsymbol{\pi_{\theta^*}} = \arg\max_{\boldsymbol{\pi_\theta} \in \Pi} \mathbb{E}_{\mathcal{D}} \left[\ln \boldsymbol{\pi_\theta}(\mathbf{a}_t^e | \mathbf{h}_t) \right], \tag{3}$$

which is similar to (2), except that the maximum likelihood learning happens under a different data distribution ($\mathcal{D}$). DAgger considers mixture policies, $\boldsymbol{\pi}^\beta = \beta \boldsymbol{\pi^e} + (1-\beta)\boldsymbol{\pi_\theta}$, for interacting with $\mathcal{M}$. $\mathcal{D}$ corresponds to the aggregated dataset of trajectories collected with the mixture policies. β is annealed to 0 over multiple iterations so that the distribution of trajectories in $\mathcal{D}$ converge to the distribution of trajectories formed by the Markov chain of $\mathcal{M}$ and $\boldsymbol{\pi_\theta}$. Algorithm 1 provides a full description of the DAgger algorithm adapted to our setting. Note that DAgger requires access to a more interactive expert that can provide the correct action for any queried state. Thus, DAgger is typically used with a computational expert (as opposed to human demonstrations), which corresponds to cases (1) and (2) described earlier for obtaining demonstrations.

Algorithm 1 DAgger for imitation learning

Input expert policy $\boldsymbol{\pi^e}$, mixing coefficient β, decay rate $\omega < 1$, and batch size N. Initialize $\mathcal{D} = \{\}$ and $\boldsymbol{\pi_\theta}$.
for $k = 1$ **to** K **do**
  Define mixture policy $\boldsymbol{\pi}^\beta = \beta\boldsymbol{\pi^e} + (1-\beta)\boldsymbol{\pi_\theta}$
  Collect dataset $\mathcal{D}_k = \{\boldsymbol{\tau}^{(1)}, \ldots \boldsymbol{\tau}^{(N)}\}$ by rolling out $\boldsymbol{\pi}^\beta$. Also collect $\mathbf{a}_t^e \sim \boldsymbol{\pi^e}(\cdot|\mathbf{s}_t)$ at every state visited in the rollouts.
  Aggregate dataset: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_k$
  Update $\boldsymbol{\pi_\theta}$ by re-solving optimization problem in (3)
  Decay the mixing coefficient: $\beta \leftarrow \omega \times \beta$
end for

IV. Sensorimotor policy architecture

Our goal is to learn sensorimotor policies with deep neural networks for complex dexterous manipulation tasks. The sensing capabilities of the agent are described in Section III-A. The policy takes as input RBG camera observations and a proprioceptive feature vector describing joint positions and velocities. The frame size of image observations is 128×128. In addition, the contact sensors detecting binary on-off contact events are fed into the policy, and helps to perform fine grained contact-rich motions. Overall, we have 21 contact sensors, 4 on each finger and 1 on the palm. Figure 2 provides an overview of the policy architecture used in this work. The deep sensorimotor policy encodes the pixel observation through a sequence of convolutional (CNN) layers, and the proprioceptive and tactile sensors through fully connected (MLP) layers. Features from all the modalities are concatenated and used to predict the robot actions.

To manipulate objects, an understanding of their velocity is required. For this purpose, we use the last three frames in our policy architecture, with each frame rendered at $20ms$ intervals. We experimented with early fusion (which combines visual information from the start), as well as late fusion, where frames are processed separately through the same CNN layers and the feature vectors are combined downstream [49]. See Figure 3a for a visual illustration. We found that late fusion works marginally better, as illustrated in Figure 3b, and use it in our experiments. This is likely because capturing intricate long-range inter-pixel dependencies across timesteps is not important for the tasks we consider. In general, POMDPs may require the use of long histories for effective control. Various architectures like LSTM [50] and transformers [51] have been developed for these purposes. However, for the tasks we consider, we found that using the three most recent frames is sufficient for the tasks we study. Integrating longer range information did not improve the asymptotic performance.

V. Results and Discussion

In our experimental evaluation, we aim to address the following questions.

1) Can visual imitation learning provide successful policies for complex dexterous manipulation tasks using high-dimensional visual observation spaces?
2) What is the performance difference between direct RL in the visual space, and imitation learning with DAgger and BC?
3) Does the use of contact sensors provide additional gains over visual inputs, and how does this change with the degree of occlusion?

Obtaining the expert policies: To obtain demonstrations, we train a computational expert policy using a compact state representation. Concretely, we take the final trained policies from our prior work [1] as the expert policy. These policies were trained using the DAPG algorithm [1] which combines RL (specifically natural policy gradient [37], [40]) with a small number of human demonstrations obtained using virtual reality. RL and the demonstrations are combined using an additional trajectory tracking reward that encourages the learned policy to stay close to the human demonstrations. We refer readers to our prior work for additional details above this procedure, and focus on imitation learning with these compact expert policies in this work.

Figures 4 and 5 presents the learning curves comparing different algorithms and architectures. We find that successful policies can be trained with just a few hundred trajectories using imitation learning, in under an hour on a single workstation with 4 GPUs.

Algorithm efficiency: Figure 4 depicts the learning curve that compares RL in the visual space, BC, and DAgger. Firstly, we find that RL in the visual space does not make any progress in the time frame we considered, whereas the imitation learning approaches are able to train successful policies. In the case of direct training in the real world, time taken to execute a trajectory using the expert policy (including human demonstrations) and an exploratory RL policy are comparable. Thus, with imitation learning, visuomotor policies can be trained in the real world orders of magnitude faster than pure RL in the visual space. Despite simulations being much faster than real-time, the long training times are a bottleneck even for simulation to real transfer approaches, often requiring thousands of CPU hours [2] even to learn policies with compact state representations. In contrast, imitation learning allows for training of deep visuomotor policies in under an hour on a single workstation.

Comparing the imitation learning algorithms, we find that DAgger performs marginally better than BC for all tasks except the pen task, where the asymptotic success rate improves from 70% to 85%. DAgger provides corrective actions by interactively rolling out the policy. These interactive corrections are particularly useful for tasks where precision is important, such as the pen repositioning task. Although DAgger performs better, the good performance of BC is encouraging. BC can be applied with human demonstration data, and does not require access to a computational expert policy. This opens an avenue for training visuomotor policies even with cheap infrastructure and laboratory setup.

Benefits of the added touch sensing: We compare the performances of the policy with and without contact sensors in Figure 5. We observe that the contact sensors provide substantial gains for the object relocation task, improving the

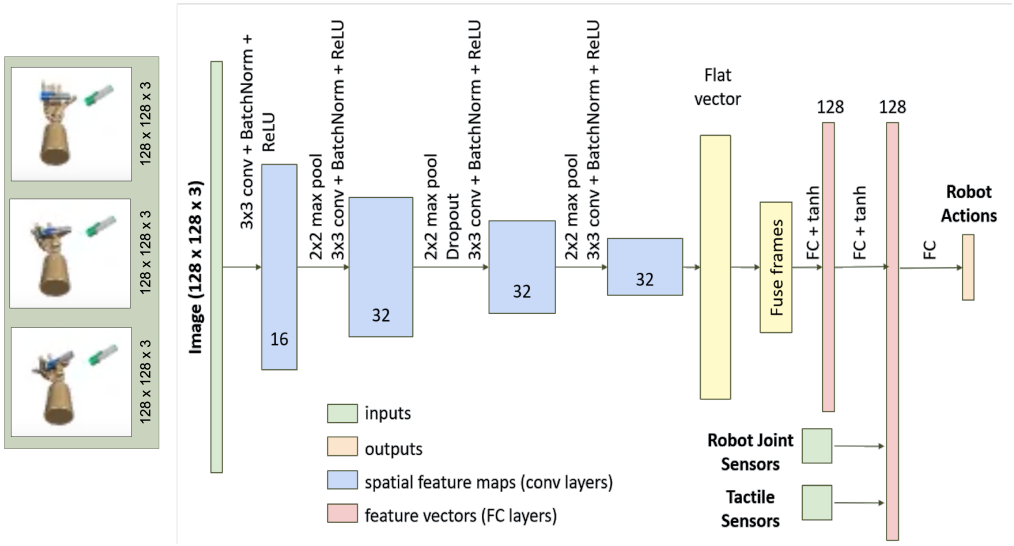
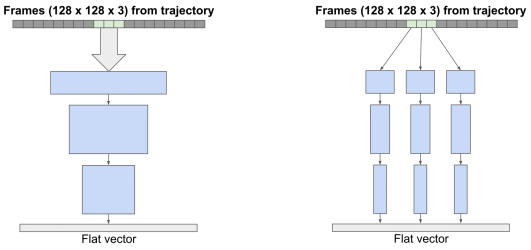
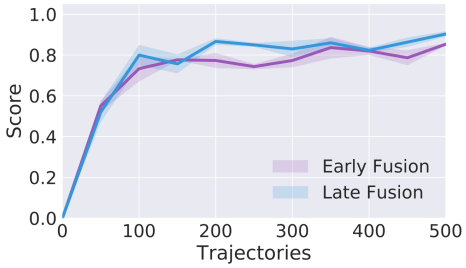


Fig. 2: Architecture of deep sensorimotor policy network. The RGB input frame is passed through a sequence of 3×3 convolutional filters interleaved with max pooling. Each layer after the first has 32 feature maps and leads to successively reduced resolutions and increased receptive field. A late fusion of the past 3 frames is used to obtain a flat feature vector that summarizes all the visual information for the decision making part of the network. To this feature vector, the proprioceptive features and touch sensors are concatenated, to obtain an overall feature vector, which is passed through a fully connected layer. Finally another fully connected layer maps these features to the action space of the robot.

success rate from 76% to 83%. The object is significantly occluded when the hand is close to the object, and the ability of contact sensors to detect the presence of contact allows for learning a better policy. For the other tasks, contact information provides a small gain.



(a) Early (left) vs late (right) fusion architecture.



(b) Learning curve on pen repositioning task.

Fig. 3: We observed that late fusion performed marginally better and used it for our policy architecture

Varying levels of occlusion: We consider multiple viewpoints for the pen task, which result in different levels of occlusions, and study the influence of contact sensors on the policy performance. Figure 6 depicts the different views we consider, and Table I summarizes the asymptotic success rate for policies with and without contact sensors. Overall, we observe that for views with many occlusions (views 3, 4, and 5), the use of contact sensors accelerate learning and lead to better asymptotic performance. In particular, for view 5, the asymptotic success percentage improved from 76% to 84%. This suggests that contact sensors provide rich source of information about the world, which are complementary to the information from the visual sensors.

We also note that the performance gain of using contact sensors appear to be small largely due to the policy with only vision (and propreoception) performing very well. We conjecture that learning based approaches are able to use patterns in the visual observations (and corresponding learned representations) to infer details that are not directly present in the observations. For example, if there is a high degree of consistent correlation between certain poses of the hand and contact events, then learning based approaches may be able to extract a large fraction of information from the touch sensing modality without any contact sensors. Significant errors in proprecoception may reduce these correlations and we conjecture that the contact sensors would provide significant additional benifit in such cases, and would make for interesting future work.

VI. Conclusions

In this work, we studied approaches to train visuomotor policies for a suite of complex dexterous manipulation tasks.

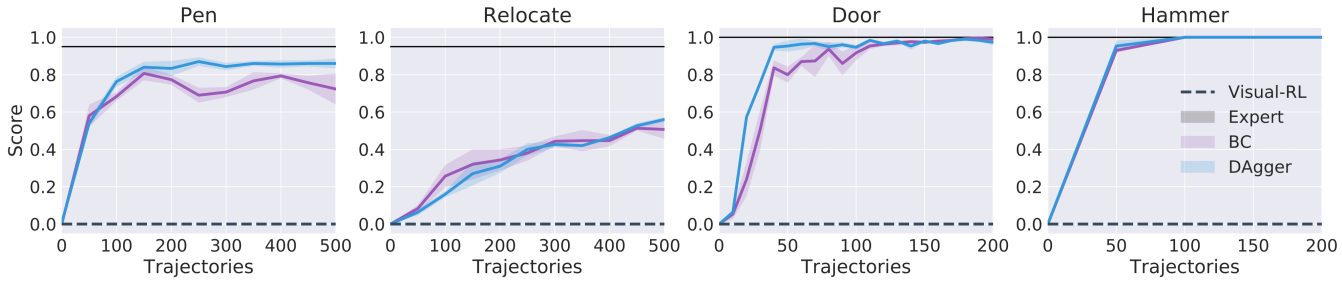
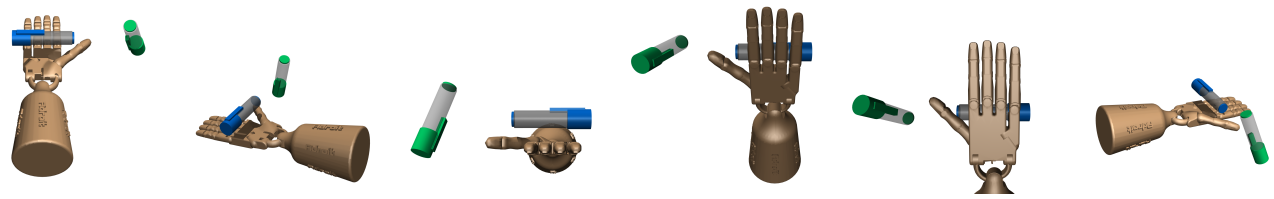


Fig. 4: Learning curves (3 random seeds) for the hand manipulation tasks. We find that direct RL in the visual space (Visual-RL) is very inefficient, and that DAgger trains marginally faster and leads to better asymptotic performance.

Fig. 5: Learning curves (3 random seeds) for the hand manipulation tasks comparing policies with (a) visual and proprioceptive inputs only (W/O Tactile) vs (b) visual, proprioceptive, and tactile inputs (With Tactile). The use of tactile sensing allows for faster learning in all the tasks, with big gains in the object relocation and hammering tasks. The viewpoints used for these tasks, as depicted in Figure 1, lead to large occlusions making the grasping harder. The use of contact information allows to easily discern the success of the grasp, enabling faster learning.



(a) View 1 (b) View 2 (c) View 3 (d) View 4 (e) View 5 (f) View 6

Fig. 6: Different viewpoints for the pen repositioning task to study effectiveness of tactile sensing. View 1, 2, and 6 have minimal occlusions whereas we observe occlusions in Views 3, 4, and 5. Correspondingly, we see larger differences in learning speed and asymptotic performance for views with larger occlusions (illustrated in Table 1).

TABLE I: Asymptotic success Percentage for the different viewpoints in the pen repositioning tasks. CS stands for contact sensor. Due to more occlusions in viewpoints 3, 4, and 5, we see a larger difference between the final performances. Results are averaged over 3 random seeds.

Sensing \ Views	1	2	3	4	5	6
Without CS	88	91	88	83	76	91
With CS	89	93	93	87	84	93

While end to end learning with RL was found to be inefficient, we surprisingly found that complex visuomotor policies can be trained with just a few hundred expert demonstrations, and in under one hour on a single workstation. This suggests that imitation learning could be a viable approach for robots to quickly learn a large repertoire of skills. In addition, we observed that using contact sensors accelerates the learning process and leads to better asymptotic performance. The difference in performance is especially noticeable for tasks and viewpoints with many occlusions.

ACKNOWLEDGEMENTS

This work was initially completed as a course project for the deep reinforcement learning course at the University of Washington by DJ, AL, and SS, under the supervision of AR and VK. Results were interpreted by AR, VK, and ET, who also developed the exposition for the paper. The authors thank Kendall Lowrey and Svetoslav Kolev for valuable discussions about the project.

REFERENCES

[1] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[2] OpenAI, "Learning dexterous in-hand manipulation," *CoRR*, vol. abs/1808.00177, 2018.

[3] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control," in *ICLR*, 2019.

[4] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, "EPOpt: Learning Robust Neural Network Policies Using Model Ensembles," in *ICLR*, 2017.

[5] K. Lowrey, S. Kolev, J. Dao, A. Rajeswaran, and E. Todorov, "Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system," *CoRR*, vol. abs/1803.10371, 2018.

[6] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *CoRR*, vol. abs/1710.06542, 2017.

[7] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *CoRR*, vol. abs/1804.10332, 2018.

[8] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *I. J. Robotics Res.*, vol. 35, no. 1-3, pp. 161–185, 2016.

[9] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *IROS*, 2016.

[10] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics Association, 2012, pp. 137–144.

[11] V. Kumar, Y. Tassa, T. Erez, and E. Todorov, "Real-time behaviour synthesis for dynamic hand-manipulation," in *ICRA*, 2014.

[12] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *I. J. Robotics Res.*, vol. 33, no. 1, pp. 69–81, 2014.

[13] I. Mordatch, K. Lowrey, and E.Todorov, "Ensemble-CIO: Full-body dynamic motion planning that transfers to physical humanoids," in *IROS*, 2015.

[14] I. M. Bullock, R. R. Ma, and A. M. Dollar, "A hand-centric classification of human and robot dexterous manipulation," *IEEE transactions on Haptics*, vol. 6, no. 2, pp. 129–144, 2013.

[15] X. Zhu and J. Wang, "Synthesis of force-closure grasps on 3-d objects based on the q distance," *IEEE Transactions on robotics and Automation*, vol. 19, no. 4, pp. 669–679, 2003.

[16] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, 2004.

[17] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[18] H. van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *Humanoids*, 2015.

[19] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, no. 1-2, pp. 171–203, 2011.

[20] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, and V. Kumar, "Dexterous Manipulation with Deep Reinforcement Learning: Efficient, General, and Low-Cost," *CoRR*, vol. abs/1810.06045, 2018.

[21] Y. Chebotar, O. Kroemer, and J. Peters, "Learning robot tactile sensing for object manipulation," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3368–3375, 2014.

[22] B. Amos, L. Dinh, S. Cabi, T. Rothörl, S. G. Colmenarejo, A. Muldal, T. Erez, Y. Tassa, N. de Freitas, and M. Denil, "Learning awareness models," *CoRR*, vol. abs/1804.06318, 2018.

[23] M. Mudigonda, P. Agrawal, M. R. DeWeese, and J. Malik, "Investigating deep reinforcement learning for grasping objects with an anthropomorphic hand," 2018.

[24] F. Sadeghi and S. Levine, "(CAD)2RL: Real Single-Image Flight without a Single Real Image," *ArXiv e-prints*, 2016.

[25] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *ArXiv e-prints*, 2017.

[26] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *CoRL*, 2017.

[27] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *CoRR*, vol. abs/1603.02199, 2016.

[28] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413, 2016.

[29] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Y. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," *CoRR*, vol. abs/1710.04615, 2017.

[30] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," *CoRR*, vol. abs/1709.10089, 2017.

[31] Y. Zhu, Z. Wang, J. Merel, A. A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, and N. Heess, "Reinforcement and imitation learning for diverse visuomotor skills," *CoRR*, vol. abs/1802.09564, 2018.

[32] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn, "Universal planning networks," *CoRR*, vol. abs/1804.00645, 2018.

[33] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, and S. Levine, "Path integral guided policy search," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3381–3388, 2017.

[34] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *International Conference on Intelligent Robots and Systems*, 2012.

[35] V. Kumar, "Manipulators and manipulation in high dimensional spaces," Ph.D. dissertation, University of Washington, Seattle, 2016.

[36] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[37] S. M. Kakade, "A natural policy gradient," in *Advances in neural information processing systems*, 2002, pp. 1531–1538.

[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[39] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *International Conference on Learning Representations (ICLR2016)*, 2016.

[40] A. Rajeswaran, K. Lowrey, E. Todorov, and S. Kakade, "Towards Generalization and Simplicity in Continuous Control," in *NIPS*, 2017.

[41] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations (ICLR2016)*, 2016.

[42] S. Kakade, "On the sample complexity of reinforcement learning," Ph.D. dissertation, University College London, 2003.

[43] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *ICRA*, 2018.

[44] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. A. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," *CoRR*, vol. abs/1707.02286, 2017.

[45] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," *IROS*, 2012.

[46] M. Al Borno, M. de Lasa, and A. Hertzmann, "Trajectory Optimization for Full-Body Movements with Complex Contacts," *IEEE Transactions on Visualization and Computer Graphics*, 2013.

[47] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," in *ACM SIGGRAPH*, 2018.

[48] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *AISTATS*, 2011.

[49] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.